
Subject: OpenMP

Posted by [281264](#) on Wed, 29 Dec 2010 09:00:33 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi,

I have been testing OpenMP with the MSC9 compiler in simple examples; it works in non debugging option; the debugging is not possible due to the non availability of the file vcomp90d.dll; I reckon the file is not included in the free version of the MSC. Any suggestion?

On the other hand, I wonder whether in U++ there are functions (apart from the thread et al.) that do -to a certain extent- similar tasks: i.e. CoWork.

Have you guys experience using OpenMP in U++?. Any hint/suggestion?.

Cheers,

Javier

Subject: Re: OpenMP

Posted by [dolik.rce](#) on Wed, 29 Dec 2010 11:23:40 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi Javier,

I am using OpenMP on Linux without any problems, but I never tried on windows. Maybe you could try with TDM mingw. r just have a look at the internet, I am pretty sure you could find the required dll for download somewhere - just remember to ask your lawyer if it is legal to download it

Also some quick googling revealed a post saying that using `#include <omp.h>` instead of `#include <openmp.h>` solves this situation, but as I am Linux only I can't confirm that

Best regards,
Honza

Subject: Re: OpenMP

Posted by [281264](#) on Wed, 29 Dec 2010 11:36:01 GMT

[View Forum Message](#) <> [Reply to Message](#)

Honza,

Thanks. Yes, MinGw/GCC works fine with OpenMP; the problem is not the inclusion of omp.h is the mentioned file.

Any feedback on the performance/usability of OpenMP? Any piece of advice? Do you also use other tools?

Cheers,

Javier

Subject: Re: OpenMP
Posted by [dolik.rce](#) on Wed, 29 Dec 2010 13:48:50 GMT
[View Forum Message](#) <> [Reply to Message](#)

281264 wrote on Wed, 29 December 2010 12:36
Any feedback on the performance/usability of OpenMP? Any piece of advice? Do you also use other tools?
OpenMP is IMHO a very handy tool if you are lazy and want to speed up some heavy computation. By lazy I mean you don't want to mess up with worker threads and all the related stuff. As long as there are some long for cycles which don't need to be processed in order, splitting them using OpenMP gives you great results for very little work. The only trouble is when you come to a point when you need to scale your app up and run it on cluster - then you have to rewrite it using other tools, most probably mpi.

I am using OpenMP for tedious tasks like counting distances between couples of atoms in atomic particles. The performance gain using omp is quite adequate. The thing I like most about OpenMP is that the code can be very simply compiled without it (basically just switching one flag to exclude the gomp library). That is great for debugging sometimes.

Honza

Subject: Re: OpenMP
Posted by [mirek](#) on Wed, 29 Dec 2010 14:40:35 GMT
[View Forum Message](#) <> [Reply to Message](#)

Quick peek at OpenMP - it looks like relying on pragmas to tell compiler how paralellize loops?

BTW, tried CoWork?

Mirek

Subject: Re: OpenMP
Posted by [dolik.rce](#) on Wed, 29 Dec 2010 16:10:28 GMT
[View Forum Message](#) <> [Reply to Message](#)

mirek wrote on Wed, 29 December 2010 15:40 Quick peek at OpenMP - it looks like relying on pragmas to tell compiler how parallelize loops? Yes, but not only loop parallelization. It allows also to work with barriers, critical sections, thread local storage, ordered sections in parallel code, atomic operations, reduction, ...

There is also few functions to work with locking, timing and scheduling. Most of those can be found in U++, in some form.

Nice overview can be found in this PDF summary.

mirek wrote on Wed, 29 December 2010 15:40 BTW, tried CoWork?

Yes, I tried it. It is a nice tool, but OpenMP lets you do slightly more in a little easier way. But the biggest difference IMHO is that OpenMP pragmas in code are much easier to read than the Cowork with its callbacks.

Honza